

# INDOOR POSITIONING USING WLAN FINGERPRINT MATCHING AND PATH ASSESSMENT WITH RETROACTIVE ADJUSTMENT ON MOBILE DEVICES

Eros Gulo<sup>1</sup>, Gunho Sohn<sup>1</sup>, Afnan Ahmad<sup>1</sup>

<sup>1</sup> Earth and Space Science and Engineering, York University, Toronto, Canada - (erosgulo, gsohn, aafnan)@yorku.ca

Commission IV, WG IV/4

**KEY WORDS:** Indoor Positioning, Fingerprint, Path Assessment, WLAN localization, Location

## ABSTRACT:

With the increasing number and usage of mobile devices in people's daily life, indoor positioning has attracted a lot attention from both academia and industry for the purpose of providing location-aware services. This work proposes an indoor positioning system, primarily based on WLAN fingerprint matching, that includes various minor improvements to improve the positioning accuracy of the algorithm, as well as improve the quality and reduce the collection time of the reference fingerprints. In addition, a novel Path Evaluation and Retroactive Adjustment module is employed; it intends to improve the positioning accuracy of the system in a similar fashion to a Pedestrian Dead Reckoning implemented along with WLAN Fingerprint Matching in a Sensor Fusion system. The benefit of this approach being that it avoids the requirement of inertial sensor data, as well as its intensive computation and power use, while providing a similar accuracy improvement to Pedestrian Dead Reckoning. Our experimental results demonstrate that this may be a viable approach for positioning using mobile devices in an indoor environment.

## 1. INTRODUCTION

### 1.1 Introduction

Indoor positioning and navigation is a burgeoning field in the greater umbrella of location-based services that has hit its stride in recent years along with and primarily because of the ubiquity of smartphones. The major area of interest in the last few years has been the application of different indoor positioning methods on the smartphone (and tablet) platform. In addition, GPS positioning and navigation has been very successful and dominant in the context of outdoor positioning on mobile devices. This, in turn, has resulted in users demanding the same level of service in the indoor space.

Due to the inherently smaller indoor spaces in relation to outdoor spaces, acceptable accuracy in indoors is 1 – 5 m, as opposed to the 5 – 15 m acceptable accuracy in outdoors. In addition, there is no set-up cost for GPS, just a receiver and positioning software built into the mobile device. This results in very high expectations set for indoor positioning solutions; an ideal solution should be sufficiently accurate, have low or no setup and maintenance costs, and should be a universal solution that can be used by most people (i.e. anyone with a smartphone) in most places. This research looks to build on previous work and approaches to advance towards an ideal solution.

We consider the various approaches applicable on the smartphone platform, and examine the particular benefits and limitations of the platform. We posit a positioning system, primarily based on WLAN fingerprint matching, that includes various minor improvements to improve the positioning accuracy of the algorithm, as well as improve the quality and reduce the collection time of the reference fingerprints. Finally, we employ a novel Path Evaluation and Retroactive Adjustment module intended to improve the positioning accuracy of the system in a similar fashion to a Pedestrian Dead Reckoning implemented along with WLAN Fingerprint Matching in a Sensor Fusion system. The benefit of this

approach being that it avoids the requirement of inertial sensor data, as well as its intensive computation and power use, while providing a similar accuracy improvement to Pedestrian Dead Reckoning.

## 2. BACKGROUND

### 2.1 Mobile Devices

**2.1.1 Sensors:** Due to the multitude of sensors now offered by mobile devices, there is a vast amount of data that becomes available to exploit for the purpose of positioning (Lane et al., 2010). Some of the sensors that may be available in a standard smartphone are: WLAN Radio, Bluetooth Radio, Cellular Radio, Accelerometer, Gyroscope, Magnetometer, NFC, Barometer, Camera, Ambient Light and Proximity.

**2.1.2 Computing Capabilities** Today's smartphones and tablets can perform nearly the same computations as low-end laptops. Furthermore, for at least three years, mobile devices have had quad-core processors and more recently there are even some octa-core devices appearing on the market. Along with the increase in performance that is possible by an increasing number of cores, energy efficiency has also been a concern of manufacturers. Due to the trend of progressively larger screens and thinner bodies on smartphones along with faster (and more power-hungry) data transmission technologies such as LTE radios, energy on mobile devices always seems to be on short supply. Thus, CPUs have seen great power efficiency improvements along with the performance improvements.

### 2.2 Location Models

**2.2.1 Discrete:** A discrete location model means that the navigable space within the floor or building is partitioned into discrete generally equal spaces. Each individual space is then assigned a unique identifier and its location is generally set as the

location of its geometric center. When the positioning algorithm estimates a position for the subject in this location model, it chooses one or more of the discrete spaces as the most likely location(s) of the subject. This type of location model was utilized in the implemented algorithms discussed in this paper.

**2.2.2 Continuous** A continuous location model means that no partitioning of the navigable space occurs, the position of a subject can be estimated at any location in the navigable space.

This type of location model is generally used with trilateration-based positioning systems, or any other algorithms that are not limited to estimating only the best candidate out of a discrete set.

### 2.3 Location Estimation Methods

This opens the door to vast amount of existing indoor positioning methods, and even inspires a few new methods (Subbu et al., 2014). The following is a list of the most popular approaches to indoor positioning on a smartphone:

- WLAN Fingerprint Matching
- WLAN Ranging
- Bluetooth Ranging
- Pedestrian Dead Reckoning
- Magnetic Fingerprint Matching
- Image Recognition / Visual Odometry
- Hybrid / Sensor Fusion

The WLAN and Pedestrian Dead Reckoning approaches were significantly studied prior to the advent of smartphones, whereas the others were only studied in significant detail once smartphones arrived with an increasing variety of sensors. Furthermore, on the mobile platform it has become increasingly common for new research to focus on multi-sensor hybrid approaches, usually cooperatively employing various algorithms using different sensor data.

**2.3.1 Trilateration (WLAN Ranging)** WLAN signal-based approaches can be broadly categorized into two major categories, ranging and fingerprinting. Ranging methods use signal path loss models of varying complexity to estimate the distances to WLAN access points (APs) based on the received signal strengths of the APs. Trilateration is then used to position the user based on the computed distances to the various APs. These methods generally require detailed floor plans of the building and the locations of the APs in the building (Torteeke et al., 2014).

Bluetooth ranging works in a manner very similar to WLAN ranging. The main difference is that Bluetooth signals are used as opposed to WLAN. This method requires Bluetooth beacons that constantly emit an identifying signal. The user's device then computes the distances to the sensed beacons and trilateration is used to estimate a position for the user. Another key difference with WLAN ranging is that the number of beacons required for equivalent performance between the two methods is significantly higher than the number of APs required because Bluetooth beacons have a much weaker signal and therefore, range due to the importance given to low cost and energy consumption (Martin et al., 2014).

**2.3.2 WLAN Fingerprint Matching** On the other hand, fingerprinting methods require AP signal strength maps of the building, or in practical terms, a database of the expected signal strengths of APs at discrete locations in the building (reference fingerprint database); the density of these discrete locations may vary. The user takes a sample of the signal strengths of APs within range (live fingerprint) and various mathematical algorithms can then be used to match the live fingerprint to the most similar location in the database with respect to the signal strengths of the sampled APs (Honkavirta et al., 2009), (Farshad et al., 2013).

**2.3.3 Pedestrian Dead Reckoning** The pedestrian dead reckoning (PDR) algorithm has three components, step detection, step length estimation, and heading estimation. Steps are detected from the periodic pattern of accelerometer readings while the user is walking; step length is estimated in real-time from the amplitude of the accelerometer readings or based on user characteristics; the user's heading is determined by magnetometer readings and the gyroscope can be used to detect and correct erroneous magnetometer readings from magnetic anomalies due to the physical environment. This method essentially estimates the user's movement based on the inertial sensor measurements provided by the mobile device. The starting location, however, needs to be provided by an external source for this system, or alternatively, the user may be located after a traversing a path sufficiently long and unique to be matched to a likely potential path in the floor plan (Harle, 2013), (Durrant-Whyte and Bailey, 2006), (Bailey and Durrant-Whyte, 2006).

**2.3.4 Magnetic Anomaly Fingerprint Matching** Magnetic fingerprint matching, as the name implies, has functional similarities to WLAN fingerprint matching. Some of the materials used in the construction of larger commercial buildings, as well as large pieces of furniture within them, can create local anomalies in Earth's magnetic field measured in different areas of a building. The magnetometers available in modern smartphones are able to measure the magnetic field with enough precision to be able to distinguish local field anomalies throughout a building. Thus, a live sample of the magnetic field (fingerprint) is matched to the most similar location in a reference fingerprint database based on the magnetic field measured. Since these magnetometers measure the magnetic field in the three physical dimensions, this method is functionally equivalent to a WLAN fingerprint matching system where the signal from three APs is sensed throughout the whole building (Li et al., 2012).

**2.3.5 Image Recognition / Visual Odometry** Methods taking advantage of the mobile devices camera sensor can be grouped into two broad categories, scene recognition and visual odometry. The former refers to methods that attempt to recognize the location of the user based on photo(s) taken by their mobile device in real-time. This can be accomplished by simple image matching if a considerable database of images with known location exists; alternatively, live images can be used to construct a three-dimensional model of the scene in view and that model then matched to a particular area of the building based on an existing accurate three-dimensional model of the building's interior (Mautz and Tilch, 2011). Visual odometry, on the other hand, uses live video from the camera to estimate the user's movement. By creating and continuously updating a three-dimensional model of the view in the live video, the change in perspective over time can convey the user's movement. Visual odometry is thus very similar to PDR, with the main difference just being how the user's movement is estimated (Nister et al., 2004).

**2.3.6 Sensor Fusion** Finally, hybrid algorithms combine multiple individual positioning methods into a larger more complex positioning system. Some implementations employ as many individual algorithms as possible and then attain a final position by averaging all of the estimates, sometimes varying the influence of individual estimates depending on their expected accuracy. Other implementations choose the individual algorithms strategically so that they may reinforce each others weaknesses; a common example is a WLAN algorithm and PDR algorithm working cooperatively; the WLAN algorithm is preferred to estimate an initial location, while the PDR algorithm is generally superior in estimating the users movement relative to an initial location, and thus, is given precedence when the user is moving (Subbu et al., 2014), (Harle, 2013).

### 3. LOCATION FINGERPRINTING

Here we discuss how we prepare the positioning environment and set up the positioning system. Broadly, this involves the collection, processing and storage of the reference data.

#### 3.1 Data Collection

**3.1.1 Collection Tools** Custom Android application was created for the purpose of collecting WLAN AP fingerprints. Application interface shows a constantly updating list of visible APs, showing their SSID, RSSI, and time of the reading. The interface also contains a text box for the name of this collection instance and a button to write the collected readings to a .txt file. Furthermore, options exist to timestamp the collection, filter out non-university infrastructure APs, and to compute and record only the mean of the RSSI readings obtained for each AP. The scanning for APs can be paused/resumed, restarted, and manually refreshed if automatic scans stop occurring. (screenshots of application will follow)

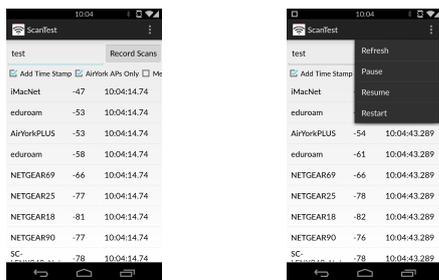


Figure 1. Main User Interface of Android based Data Collection Tool

The application was installed on multiple devices that were used for collecting fingerprints over the course of the research. Devices included the LG Nexus 4 and 5, Samsung Nexus 10, Sony Xperia Z3 Compact, and OnePlus One.

**3.1.2 Collected Data** Prior to data collection, the positioning environment must be set up. Within our work, given that a symbolic location system was used, the positioning area was set up semi-automatically. Key locations were manually chosen, then spaces between them were automatically filled by additional uniformly spaced out symbolic locations.

The reference WLAN fingerprint data can be collected once the symbolic locations are set. The Data Collection tool provides the

data captured from the scans in the following format:

- Building
  - Floor
    - Symbolic Location 001
      - AP<sub>1</sub> (MAC, SSID, number of RSSI readings)
        - RSSI<sub>1</sub>:timestamp
        - RSSI<sub>2</sub>:timestamp
        - ...
      - ...
    - Symbolic Location 002
      - AP<sub>1</sub> (MAC, SSID, number of RSSI readings)
        - RSSI<sub>1</sub>:timestamp
        - RSSI<sub>2</sub>:timestamp
        - ...
        - ...
      - ...

#### 3.2 Fingerprint Database

**3.2.1 Storage** Primarily, the fingerprint database is stored on text files; each symbolic location has one text file from each device used to collect fingerprints at that location. The format of each file is as follows:

- Symbolic Location 001 (# of APs)
  - AP<sub>1</sub> (MAC, SSID, number of RSSI readings)
    - RSSI<sub>1</sub>:timestamp
    - RSSI<sub>2</sub>:timestamp
    - ...
  - AP<sub>2</sub> (MAC, SSID, number of RSSI readings)
    - RSSI<sub>1</sub>:timestamp
    - RSSI<sub>2</sub>:timestamp
    - ...
  - ...

Once the positioning algorithm is initialized (text files are read and a live database structure is created on memory), the positioning application can create a backup txt file of the database containing only the simplified information (see section 3.2.3) in order to speed up initialization for the next time positioning is attempted on the same floor with the same initialization parameters.

**3.2.2 Filtering** The main filtering that occurs is with respect to the APs that are considered. The data collection application collects readings from all APs in range, however, the positioning algorithm only uses readings from the APs that are part of the university's WLAN infrastructure. When creating the live database, APs are filtered based on their SSID (whitelist of university infrastructure SSID's used).

Other filtering may also occur based on the initialization parameters, for example, APs with very erratic RSSI readings (beyond a certain threshold), or APs with very low mean RSSI (again, beyond a certain threshold), may be removed from the database.

Finally, filtering generally also occurs for real-time readings during positioning as well as test readings used for performance evaluation. The filtering in these cases occurs on a fingerprint-by-fingerprint basis; a fingerprint may be discarded or combined

with the following/preceding fingerprint depending on the number of APs represented in that fingerprint and/or the timestamps of RSSI readings contained in it. Occasionally, fingerprints read by the positioning algorithm are incomplete, thus, they will contain fewer RSSI readings than expected for that particular location; the remaining expected RSSI readings then come in the following fingerprint with timestamps a millisecond later than the previous fingerprint. Thus, in these cases, the two fingerprints are combined into one before being input in the positioning algorithm. In the much more rare cases where the incomplete fingerprint is not followed by its complementary incomplete fingerprint, the fingerprint is simply discarded as it is deemed unreliable. The general criterion for designating a fingerprint as incomplete is a threshold for the minimum expected AP RSSI readings for any fingerprint attained on a particular floor; this threshold is based on the number of APs visible at any given position on that floor.

**3.2.3 Simplification** In this context, simplification refers to the creation of the live fingerprint database used by the positioning algorithm in real-time. During initialization, the application reads all of the fingerprint txt files for all symbolic locations on the building and floor where the user is to be positioned. The application then creates a data structure that contains reference fingerprints for every symbolic location on that floor.

Generally, a mean value is calculated from the multiple RSSI readings for each AP in the txt files; this mean value is then added to the data structure and the individual RSSI readings are discarded to conserve memory. Depending on the initialization parameters, the data structure could also contain the number of readings as well as a measure of the variability of the signal strength for each AP at each symbolic location.

When the Bayes Maximum Likelihood algorithm is used for positioning, then normalized RSSI histograms are created for every AP at every symbolic location.

To speed up initialization on future positioning sessions, all the info retained in the live fingerprint database is written to a new file, thus, allowing future initialization with the same parameters to recreate the live fingerprint database without having to do any of the computations in the first initialization.

#### 4. LOCATION INFERENCE / ESTIMATION

##### 4.1 *k*-Nearest Neighbours

**4.1.1 *k*-Nearest Neighbours Algorithm** The *k*-Nearest Neighbours (*k*-NN) algorithm is a simple machine-learning algorithm, generally used in pattern recognition. The algorithm consists of searching through a set of reference points for the *k*-nearest points of that set to a query point in the same dimensional space. If there are multiple query points, the algorithm is repeated for each one. Nearness or proximity can be arbitrarily defined, although often Euclidean or Manhattan distance is used as a metric of proximity. Upon the identification of the *k* nearest reference points, those points can be used to either assign a class (*k*-NN classification) or assign a value (*k*-NN regression) to the query point.

**4.1.2 Application to WLAN Fingerprint Matching** The *k*-NN algorithm can be utilized in WLAN Signal Fingerprint Matching (WSFM), a method of positioning in (mostly indoor)

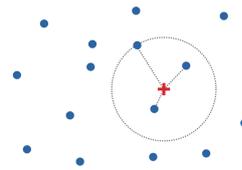


Figure 2. Diagram of *k*-Nearest Neighbours

areas with WLAN infrastructure. WSFM requires a database or map of the signal strength of all WLAN access points (APs) throughout the area where positioning is possible. The user then obtains a sample of the signal strengths of all APs (fingerprint) at their particular location, and the WSFM algorithm matches the fingerprint to the most similar location (based on the AP signal strengths) in the database.

A fingerprint typically contains the information in the table below, with the set of RSSI readings, denoted by  $RSSI^S = RSSI_1, RSSI_1, RSSI_1, \dots, RSSI_n$ , used as the input/measurement for *k*-NN.

Time Received	AP Network Name	AP Identifier	Signal Strength
$t_1$	SSID <sub>1</sub>	BSSID <sub>1</sub>	RSSI <sub>1</sub>
		BSSID <sub>2</sub>	RSSI <sub>2</sub>
		BSSID <sub>3</sub>	RSSI <sub>3</sub>
	SSID <sub>2</sub>	BSSID <sub>4</sub>	RSSI <sub>4</sub>

Table 1. Information Typically Collected in a Fingerprint

The *k*-Nearest Neighbours algorithm (*k*-NN) is used to match the subjects fingerprint to the symbolic location with the reference fingerprint most similar to it. Every location estimation, that is, whenever the subject collects a new fingerprint, *k*-NN compares the subjects fingerprint to every single symbolic location in the reference fingerprint database and returns the *k* symbolic locations that are nearest to it in terms of AP signal strengths.

When using *k*-NN, the database is set up as a set of reference points (or reference fingerprints) in signal space that correspond to particular locations in the positioning area. Thus, let the reference database be  $L = l_1, l_2, \dots, l_p$ , where *p* represents the number of specific locations represented in the database and  $l^i$  typically contains the information in the table below. Also contained in  $l_i$  is  $RSSI^R = RSSI_1, RSSI_1, RSSI_1, \dots, RSSI_m$ , the reference point in signal-space used in *k*-NN.

Symbolic Location	Geometric Location	AP Network Name	AP Identifier	Mean Signal Strength
001	$(x,y)_{001}$	SSID <sub>1</sub>	BSSID <sub>1</sub>	RSSI <sub>1</sub>
			BSSID <sub>2</sub>	RSSI <sub>2</sub>
			BSSID <sub>3</sub>	RSSI <sub>3</sub>
		SSID <sub>2</sub>	BSSID <sub>4</sub>	RSSI <sub>4</sub>

Table 2. Information restructured based on its Symbolic and Geometric Location

Finally, once all the distances are calculated between  $RSSI^S$  and every  $RSSI^R$ , they are sorted and the *k* points in *L* with the smallest distances to the fingerprint are selected to estimate a location for that fingerprint. We also experimented with a modified version of this algorithm (WSDFM) that uses a different definition of the fingerprint, namely the differences between the signal

strengths, as opposed to the absolute signal strengths, of all visible APs. Therefore, distance calculations in this version of the algorithm are in  $\frac{d(d-1)}{2}$  dimensions when  $d$  APs are visible in the fingerprint.

**4.1.3 Definition of Proximity (Distance Calculation)** The metric of proximity in my implementation is a normalized Manhattan distance; for each fingerprints  $RSSI^S$ , the distances are calculated between it and the  $RSSI^R$  of every reference fingerprint in  $L$ . The proximity metric,  $d_{ss}(l)$ , is described by the equation below.

$$d_{ss}(l) = \frac{\sum_{m=1}^{i=1} |RSSI_i^{R(l)} - RSSI_i^S| + (\beta_0(m-n))}{m} \quad (1)$$

$$L^{kmin} = \arg \underset{k=5}{kmin} d_{ss}(l), L^{kmin} \subset L \quad (2)$$

The symbolic locations corresponding to the  $k$ -smallest distances in signal space are identified.

**4.1.4 Distance Dimensions (Visible AP's)** It should be noted that  $RSSI^S$  and  $RSSI^R$  are often not defined in the same dimensions, that is to say  $n$  and  $m$ , the APs represented in  $RSSI^S$  and  $RSSI^R$ , respectively, are often not the same. Not all APs are necessarily visible in every location of the positioning area, thus, the potential mismatch of dimensions must be dealt with.

**Normalization** One way to deal with a mismatch of dimensions is to normalize the proximity metric, in this case, Manhattan distance. This allows the proximity metric to represent the average distance per dimension as opposed to the combined distance in all dimensions. A solution that relies solely on normalization will compute a distance using only the dimensions that match between  $RSSI^S$  and  $RSSI^R$ , i.e. those that correspond to the same APs. This means that non-matching dimensions are simply ignored, and this can result in outliers and generally poor accuracy when the number of matching dimensions is very low.

**Artificial RSS** Another way to deal with dimension mismatch is to insert artificial values into  $RSSI^S$  for the dimensions of  $RSSI^R$  that are not represented in  $RSSI^S$ . In this case, the value inserted ( $RSS^*$ ) is usually the lowest possible RSS value that can be sensed, often set to -100 dBm. This solution assumes that  $RSS^*$  is practically equivalent to the real RSS in that spot (which is too low to be sensed by the subjects device). Supporting rationale for this method is that by inserting a set minimum value ( $RSS^*$ ), the penal impact on the metric for that reference location will be proportional to the strength of the missing AP signal (missing dimension).

**Hybrid/Penalization for Mismatch** The two methods of dealing with dimension mismatch discussed above can be used together in a hybrid solution where the artificial  $RSS^*$  readings are inserted into the  $RSSI^S$  and then the Manhattan distance is also normalized. The rationale for the latter is that even  $RSSI^R$  of different symbolic locations will vary in the dimensions represented (APs visible), thus, normalization can still help make the proximity comparison fairer.

This hybrid method certainly improves the accuracy of positioning of the  $k$ -NN algorithm, however, during performance testing, a different hybrid method (penalization method) was also

tested and showed to further improve accuracy. The penalization method opts to add a set penalty to the Manhattan distance for every dimension mismatch as opposed to employing artificial RSS readings. This penalty is essentially an artificial difference between  $RSSI^R$  and  $RSSI^S$  in one dimension. The actual penalty was found heuristically by trying various values, and tests showed that using one constant value for the penalty resulted in better accuracy than the varying penalization resulting from inserting the artificial RSS values. This penalization method is demonstrated in the mathematical definition of our proximity metric in the  $(\beta_0(m-n))$  term. Here,  $\beta_0$  is the penalty value and  $(m-n)$  determines how many times to add the penalty to the proximity metric (i.e. how many dimensions are represented in  $RSSI^R$  but not in  $RSSI^S$ ).

**4.1.5  $k$ -NN Weighting** Whether performing  $k$ -NN classification or regression, when  $k > 1$ , there needs to be scheme to combine the input/effect of all  $k$  neighbours. In the simplest case, an equal weighted mean of the neighbours is computed. Another very common method is to take a weighted mean of the neighbours, where the weight of each neighbour is inversely proportional to its rank or  $k$ -value. Other, more complex, methods can also take into account the actual metric (e.g. Euclidean distance) of the nearest neighbours calculation for each of the neighbours to determine that neighbour's weight, or simply retain all of the neighbours and use an additional algorithm to obtain a single estimate from the  $k$  neighbours. The various methods of assigning weight to the neighbours can also be broadly categorized into two categories, apriori and aposteriori weighting; however, some complex methods can even perform both types of weighting.

**Pre-defined (apriori) Constant Weights** Apriori weights generally refer to scalar constant weights applied to the neighbours. This generally includes the trivial case, where all neighbours are equally weighted, and cases where the weights are based on the rank. Rank based weights will generally be something along the lines of  $1/\text{Rank}$ , or in other cases, each rank can have a specific arbitrarily predefined weight (i.e.  $1^{\text{st}} = 1.0$ ,  $2^{\text{nd}} = 0.8$ ,  $3^{\text{rd}} = 0.6$ , and etc.) generally based on some prior knowledge about the likelihood of each neighbour being correct.

**Calculated (aposteriori) Weights (Bayes ML)** Aposteriori weights generally refer to the determination of weights for the neighbours after the  $k$ -nearest neighbours have been determined. This can be accomplished in a number of ways. The simplest way is to use the distance metric of each neighbour that determined its proximity to the query point (e.g. the weight could be  $1/d_i$ ). A more complex way is to use an alternate algorithm to determine the weights of the neighbours. The secondary algorithm can be entirely independent of the  $k$ -NN metric, and thus, can result in weights that are not proportional to the  $k$ -NN ranks of the neighbours.

## 4.2 Path Assessment and Retroactive Adjustment

The possible paths of the subject over the last five epochs are assessed based on three criteria; these can be briefly described as  $k$ -NN score/proximity, and short-term and long-term movement regularity.

**4.2.1  $k$ -NN Proximity** Although the top  $k$  locations from  $k$ -NN are attained, this does not mean that they are equally valued. Thus, their proximity measures are also retained so that they may be considered in the  $k$ -NN proximity criterion of the path assessment stage. This criterion can be scored in three ways; the first

is relative score by comparing each of the  $k$  locations ( $l_i$ ) to the nearest location ( $l_1$ ), the second is an absolute score of the proximity, and the third is another more complex relative score.

$$kNNP(l_i^t) = \frac{d_{ss}(l_1)}{d_{ss}(l_i)} \text{ or } \frac{1}{d_{ss}(l_i)} \text{ or } \frac{d_{ss}(l_1)}{1 + d_{ss}(l_i) - d_{ss}(l_1)} \quad (3)$$

Heuristically, it was determined that the first way of scoring this criterion was the most effective.

**4.2.2 Short-Term Movement Regularity** The second criterion scored in the path evaluation is the short-term movement regularity. This criterion essentially represents how likely it is that the subject moved between two locations in the time between two epochs. This is scored by comparing the physical distance between the two locations with the distance that can be walked at an average walking speed during the time between the two location estimates (i.e. time between consecutive  $k$ -NN estimates). The score is calculated with the equation below, where  $\Delta t$  is elapsed time in seconds,  $wS$  is an average walking speed in  $m/s$ , and  $d_{PS}(l_i, l_j)$  is the Euclidean distance between the symbolic locations with  $t$  being the epoch.

$$SMR(l_i^t, l_j^{t-1}) = \frac{wS \cdot \Delta t}{d_{PS}(l_i^t, l_j^{t-1})} \quad (4)$$

Thus, the score is at a maximum when  $d_{PS}$  is less than (i.e. subject is stationary [dummy 1 cm value inserted for  $d_{PS}$  in this situation to avoid division by 0] or moving slowly) or equal to the average distance walked in the elapsed time between the consecutive epochs. On the other hand, the score decreases as the physical distance between the locations increases (once it is above the expected walked distance).

**4.2.3 Long-Term Movement Regularity** The third and final criterion scored in the path assessment is long-term movement regularity. Since the path assessment is performed for a recent portion of the subject's estimated path, namely the 5 last epochs, this criterion represents how well the subject's movement matches its net displacement over the 5 epochs. This criterion is scored by summing up the total distance travelled by the subject as they move (or stay stationary) from epoch to epoch, then comparing it to the net displacement of the subject over the 5 most recent epochs. The score is calculated with the equation below, where  $d_{PS}(l_i, l_j)$  is the Euclidean distance between the symbolic locations and  $t$  is the epoch.

$$LMR = \frac{d_{PS}(l_i^t, l_j^{t-5})}{\sum_{e=t-4}^{t-1} d_{PS}(l_i^e, l_j^{e-1})} \quad (5)$$

This score is at a maximum for this criterion when the total distance travelled is equal to the net displacement; conversely, it is at a minimum when there is a lot of movement but no net displacement. This essentially discourages a path with stuttering (i.e. one step forward, one step back, two steps forward, while subject only walked two steps forward), and tends to smooth the subjects estimated path. To make the comparison less cumbersome, the net displacement considered in the equation is direction-less; furthermore, the code implemented also automatically inserts a value of 1 for the score if the total movement (and therefore also the net displacement) is 0.

**4.2.4 Final Score** The aforementioned three scores all have different ranges and they do not follow the same curve. Thus, sigmoid functions are employed to reconcile the distributions of each of the movement regularity scores. Furthermore, the sigmoid functions allow for each of the scores to have a more precise and intended effect on the final overall score for the path segment being considered. The coefficients of the sigmoid function were heuristically determined to best attain the aforementioned behaviour. The following are the sigmoid functions used for each criterion score and their respective graphs.

$$SMR_s(l_i^t, l_j^{t-1}) = \frac{3(SMR - 0.5)}{2\sqrt{9(SMR - 0.5)^2 + 1}} + 0.5 \quad (6)$$

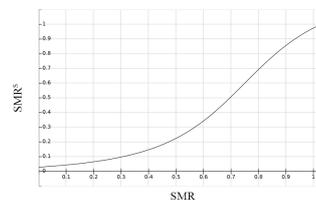


Figure 3. SMR Sigmoid Function

As can be seen in the graph, the SMR sigmoid function greatly favours situations where the subjects is moving at a normal pace (SMR is approximately 1) or staying stationary (SMR is significantly greater than 1), with the latter being slightly more favoured. As the movement increases, the score drops to discourage very fast (and likely incorrect) movement. The function outputs about half a maximum score when the subject has moved twice as far as expected and a quarter of the maximum score when the movement is three times what's expected.

On the other hand, the LMR sigmoid function does not need to deal with input LMR values greater than 1, and additionally, the score drops much faster than the SMR sigmoid function. The scoring function is intentionally more punitive for the LMR because the distances being compared in the LMR function are larger than those in the SMR function, thus, the LMR sigmoid scoring function needs to penalize even small relative differences.

$$LMR_s = \frac{15(LMR - 0.75)}{8\sqrt{9(LMR - 0.75)^2 + 1}} + 0.6 \quad (7)$$

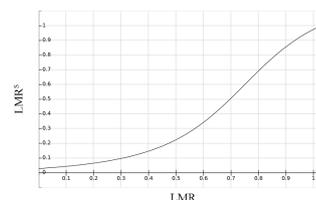


Figure 4. LMR Sigmoid Function

Once all the criteria are scored for all of the position estimates in a particular potential path, the final path score for that segment (5 epochs) is given by the following function.

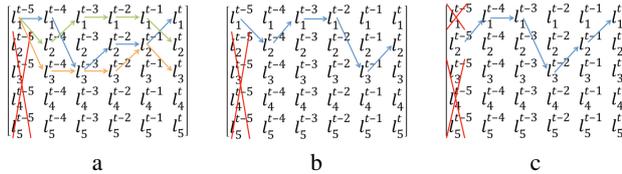


Table 3. Path Score calculation left to right

$$\begin{aligned}
 PathScore = & kNNP(l_i^{t-4}) \cdot SMR_S(l_i^{t-4}, l_j^{t-5}) \\
 & \cdot kNNP(l_i^{t-3}) \cdot SMR_S(l_i^{t-3}, l_j^{t-4}) \\
 & \cdot kNNP(l_i^{t-2}) \cdot SMR_S(l_i^{t-2}, l_j^{t-3}) \\
 & \cdot kNNP(l_i^{t-1}) \cdot SMR_S(l_i^{t-1}, l_j^{t-2}) \\
 & \cdot kNNP(l_i^t) \cdot SMR_S(l_i^t, l_j^{t-1})
 \end{aligned} \quad (8)$$

As mentioned earlier, the  $k$  chosen locations from each of the previous five epochs are retained. Thus, when we say all potential paths, we mean all possible ways to traverse the following matrix from left to right starting from the final chosen location at epoch  $t - 5(l_1^{t-5})$  in the example, Table 3 (a).

Once every possible path is scored, the path with the highest score is chosen as the best path and its final location (at  $t$ ) is presented to the subject as the subject's current position. Say, for example, that the Table 3(b) was the highest scoring path.

At the next epoch (new set of  $k$  locations provided by  $k$ -NN), shown in the matrix Table 3(c), the starting location on the left side of the matrix is  $l_2^{t-5}$ . This is the same location as  $l_2^{t-4}$  from the previous matrix, and was the only one retained from that epoch after being in the best path the last time the path was assessed.

Another way to describe the path assessment process is to think of a tree created by all the locations suggested by  $k$ -NN over the number of epochs considered. The following simplified example considers  $k$ -NN (where  $k = 3$ ) over 3 epochs. All the possible paths to go from root to leaf in the tree below are scored and then ranked.

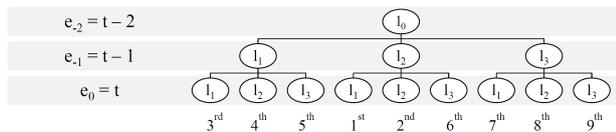


Figure 5. Path Assessment example in a tree format

The discrete locations of the highest ranked path are chosen as the subject's most likely path over the course of the epochs considered.

## 5. RESULTS

In this section, we will present results gathered from running simulation in indoor environment.

### 5.1 Experiment Setup

We conducted the experiments in three different buildings at York University. Those buildings are as follows: Petrie Science & Engineering (PSE), Chemistry Building (CB) and Bergeron Centre for Engineering Excellence (BCEE)

York University has its WLAN access points set up throughout these buildings. Despite there being a multitude of other access points in use throughout these buildings, we decided to only use the readings received from York University's network infrastructure access points. This was done because their signals were adequately available throughout the positioning environment and they provided a higher level of reliability and stability in comparison to the other WLAN access points.

Figure 6 shows distribution of York University's network infrastructure access points on the 3<sup>rd</sup> floor of the PSE building. We used collection tools described in section 3.1.1.

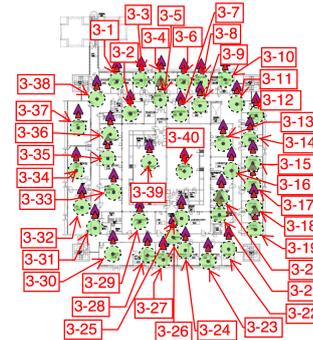


Figure 6. Distribution of AP on 3<sup>rd</sup> floor of PSE

### 5.2 Performance Evaluation

We collected thousands of positioning data points from two versions of our implemented positioning system:

1. Referred to as  $k$ -NN in this section, the first implementation was our improved WLAN Fingerprint Matching algorithm.
2. Referred to as  $k$ -NN + PERA in this section, the second implementation included the Path Evaluation and Retroactive Adjustment module in addition to the WLAN Fingerprint Matching in the first implementation.

Figures 7, 8 and 9 show the comparison between the two implementations. We observed that  $k$ -NN + PERA results in significantly improved positioning accuracy compared to  $k$ -NN. This level of positioning accuracy can certainly be useful for location-aware applications in the indoor space.

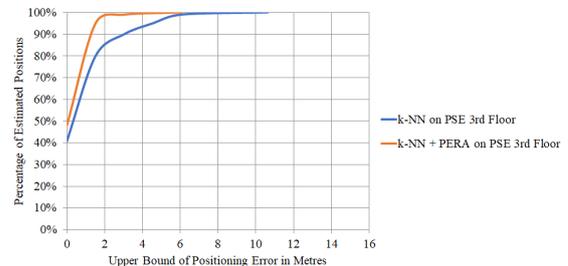


Figure 7. Cumulative Positioning Error on 3<sup>rd</sup> Floor of PSE

We see the greatest accuracy within the PSE building, positioning error less than 3 metres 90% of the time for  $k$ -NN and positioning error less than 2 metres 99% of the time for  $k$ -NN + PERA. We believe this is due to the fact that it is the oldest of the three buildings and has thick concrete walls throughout; this results

in greater heterogeneity of RSS from the various access points throughout the positioning area, therefore resulting in better performance primarily by the  $k$ -NN algorithm.

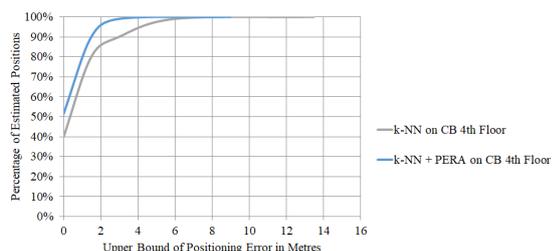


Figure 8. Cumulative Positioning Error on 4<sup>th</sup> Floor of CB

We see similar accuracy within the Chemistry building, positioning error less than 3 metres 90% of the time for  $k$ -NN and positioning error less than 2 metres 95% of the time for  $k$ -NN + PERA. Chemistry building is much newer than the PSE building, but has similar sized hallways and likely relatively similar structural materials and access point distribution.

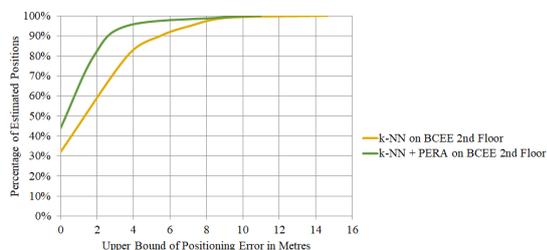


Figure 9. Cumulative Positioning Error on 2<sup>nd</sup> Floor of BCEE

In BCEE, we observed lower accuracy compared to other two buildings, positioning error less than 5 metres about 90% of the time for  $k$ -NN and positioning error less than 3 metres 95% of the time for  $k$ -NN + PERA. The most likely reason for the poorer performance in this building is the fact that it is a brand new building with large open hallways and a general open concept style. This results in more homogeneity in the RSS of the access points in large swaths of the positioning area; in open areas where the signal can travel unimpeded, it becomes difficult for WLAN Fingerprint Matching algorithms, in general, to differentiate between adjacent locations.

## 6. CONCLUSION

In this work, we presented a positioning system, primarily based on WLAN fingerprint matching, that includes various minor improvements to improve the positioning accuracy of the algorithm, we also employ a novel Path Evaluation and Retroactive Adjustment module intended to improve the positioning accuracy of the system in a similar fashion to a Pedestrian Dead Reckoning implemented along with WLAN Fingerprint Matching in a Sensor Fusion system. As can be seen in the experimental results, the WLAN fingerprint matching algorithm, referred in the previous section as  $k$ -NN, has respectable performance in its own right, achieving room-level accuracy (less than 3 - 5 metre error) 90% of the time, despite all positioning experiments taking place in open hallways. Furthermore, we show that with the PERA module, positioning accuracy is improved to less than 2 - 3 metre error about 95% of the time. In our opinion, this level of accuracy is

adequate for indoor positioning on the smartphone platform and comes without the added energy and computation cost of using the inertial sensors in a PDR algorithm. This sort of performance can also be considered practically equivalent, in our opinion, to the performance of GPS positioning on smartphones in outdoor use cases. In future research, we will look into improving the weak points of our system, namely poorer performance in larger and more open spaces, as well as making the PERA module more elaborate and context-aware to aid in cases of more unusual user movement. In addition, we hope to develop a reliable method for floor detection, which can be a difficult task in its own right within the indoor positioning realm.

## REFERENCES

- Bailey, T. and Durrant-Whyte, H., 2006. Simultaneous localization and mapping (slam): part ii. *IEEE Robotics Automation Magazine* 13(3), pp. 108–117.
- Durrant-Whyte, H. and Bailey, T., 2006. Simultaneous localization and mapping: part i. *IEEE Robotics Automation Magazine* 13(2), pp. 99–110.
- Farshad, A., Li, J., Marina, M. K. and Garcia, F. J., 2013. A microscopic look at wifi fingerprinting for indoor mobile phone localization in diverse environments. In: *International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–10.
- Harle, R., 2013. A survey of indoor inertial positioning systems for pedestrians. *IEEE Communications Surveys Tutorials* 15(3), pp. 1281–1293.
- Honkavirta, V., Perala, T., Ali-Loytty, S. and Piche, R., 2009. A comparative survey of wlan location fingerprinting methods. In: *2009 6th Workshop on Positioning, Navigation and Communication*, pp. 243–251.
- Lane, N., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T. and Campbell, A., 2010. A survey of mobile phone sensing. *Communications Magazine, IEEE* 48(9), pp. 140–150.
- Li, B., Gallagher, T., Dempster, A. G. and Rizos, C., 2012. How feasible is the use of magnetic field alone for indoor positioning? In: *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–9.
- Martin, P., Ho, B.-J., Grupen, N., Muñoz, S. and Srivastava, M., 2014. An ibeacon primer for indoor localization: Demo abstract. In: *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings, BuildSys '14*, ACM, New York, NY, USA, pp. 190–191.
- Mautz, R. and Tilch, S., 2011. Survey of optical indoor positioning systems. In: *2011 International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–7.
- Nister, D., Naroditsky, O. and Bergen, J., 2004. Visual odometry. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, Vol. 1, pp. I-652–I-659 Vol.1.
- Subbu, K. P., Zhang, C., Luo, J. and Vasilakos, A. V., 2014. Analysis and status quo of smartphone-based indoor localization systems. *IEEE Wireless Communications* 21(4), pp. 106–112.
- Torteeka, P., Chundi, X. and Dongkai, Y., 2014. Hybrid technique for indoor positioning system based on wi-fi received signal strength indication. In: *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 48–57.